UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/876,366	06/07/2001	Joseph Franklin Garvey	RAL920000123US1	9154

			EXAMINER
45503	7590	11/30/2004	STEELMAN, MARY J

DILLON & YUDELL LLP
8911 N. CAPITAL OF TEXAS HWY.,
SUITE 2110
AUSTIN, TX 78759

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 11/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/876,366	GARVEY, JOSEPH FRANKLIN
	Examiner	Art Unit	
	Mary J. Steelman	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *17 August 2004*.

2a) ☒ This action is **FINAL.**   2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
  closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *8-23* is/are pending in the application.

  4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *8-23* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

  Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

  Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

  a) ☐ All b) ☐ Some * c) ☐ None of:

  1. ☐ Certified copies of the priority documents have been received.

  2. ☐ Certified copies of the priority documents have been received in Application No. _____.

  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage
    application from the International Bureau (PCT Rule 17.2(a)).

  * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
  Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
  Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

U.S. Patent and Trademark Office
PTOL-326 (Rev. 1-04)   Office Action Summary   Part of Paper No./Mail Date 11222004

DETAILED ACTION

1.   This Office Action is in response to Amendment and Remarks received 17

August 2004. Per Applicant's request, claims 1-7 have been cancelled. Claim 23 has been

added. Claims 8-23 are pending.

Claim Objections

2.   In view of the Amendments the prior claim objections are hereby withdrawn.

Claim Rejections - 35 USC § 101

3.   In view of the Amendments the prior 35 USC 101 rejections are hereby withdrawn.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 8-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent

4,099,230 to Mead.

Mead disclosed (Abstract) a method and means for implementing the control structure of

a computer comprising…conditional execution, and nesting… Col. 2, lines 23-26, "…control

system for a computer wherein a very small number of very high level control constructs (IF,

ELSE, ELSEIF, etc.)are capable of providing the entire control structure of a very large system."

Col. 2, line 34, "…programs and their labels are loaded…", Col. 2, lines 39-45, "When a label

for a set of instructions, constituting a specific program segment, is detected, the label table is

instructed to store, at the address represented by the label, the address of the particular program segment which is derived from the program counter...whenever it is desired to call out a new set of instructions, this is done by using the label as an address ..." Col. 2, lines 50-52, "Provision is made for conditional execution and branching as well as nesting, using labels in place of addresses." Conditional constructs direct execution to labels.

Mead disclosed, col. 3, lines 22-28, "A computer will contain...a program memory module for providing the program for instructing the data processing and a state machine controller. In accordance with this invention the state machine controller includes a program tracker circuit, and a micro program control circuit."

Mead disclosed, col. 3, lines 32-49, "The program tracker circuit...comprises an arrangement for keeping track of locations within programs to which a return should be made after a branch has been executed, and also keeps track of nesting information whereby the proper return from a nested program segment through several successive layers of program segments to the authority program may be made."

Mead disclosed, col. 5, lines 27-30, "A mechanism which enables the call of a procedure defined at any point in a program from any other point in a program is the label mechanism."

Mead disclosed, col. 5, lines 46-50, "A structured program is by definition nothing more than a large set of procedures all being called from one another and being initiated from one master procedure which is called the main program."

Per claims 8 and 15:

A state machine for an assembler capable of processing structured assembly language, said state machine comprising:

-an IF state, an ELSE state, an END_IF state, an ELSE_ IF and a SETUP_IF state;

(Mead disclosed the IF and ENDIF instructions (col. 6, line 39-col. 7, line 4), the ELSE instruction (col. 7, lines 5-34), the ELSEIF instruction (col. 7, line 35-col. 8, line 31) and nesting (col. 9, lines 40-60). The state machine / micro program control capable of processing structured assembly language is disclosed at col. 14, lines 22-33, "During the run mode, when the program control sees an IF statement...the comparator compares...This condition is then written into the label stack memory...along with the label corresponding to the IF statement. This condition is used by the micro program control to determine whether or not the condition for a particular IF . or ELSEIF statement has been successfully met.")

-means for transitioning from said IF state or said ELSE_IF state to said SETUP_IF state, in response to recognizing a SETUP_IF clause;

(Mead disclosed a mean for transitioning from one state to another, col. 14, lines 41- col. 15, line 20. Mead disclosed, col. 18, lines 46-52, "...at the end of the scan interval, because of the original organization of the program segments, the program which is stored in the program memory consists of a set of program segments each beginning with a label statement and a binary label and ending with a return instruction and consisting of executionable code." Mead did not disclose a SETUP_IF clause / programming construct. However, this is merely another reserved term related to a label, created by the programmer, that the controller will recognize when running the code. Mead disclosed, col. 17, lines 65-67, "It is also clear that any other embodiment of a sequential state machine can be used to activate the microprogram sequencing.

-means for transitioning from said SETUP_IF state to said ELSE_IF state, in response to recognizing an ELSE_IF clause.

As noted above, the means for transitioning from one state to another is via the controller encountering a structured language term and referring to the label for executable code.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Mead's invention to include a SETUP_IF construct for the purpose of directing control flow, because it merely is a name that references executable code, as written by a programmer, for the purpose of logically organizing a program. Whatever the control construct is named, depending on the result of the comparison, it references the label table to find related code. Official notice is given that a SETUP_IF clause is merely a label,

defined by a programmer to define a location for an executing program to jump to, a control

construct, when certain conditions are met. When the requirements on one state are met the

program may be coded to jump to another following state.

Per claims 2, 9, and 16:

-state machine further includes a means for transitioning from said IF state to said ELSE state, in

response to recognizing an ELSE clause.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claims 3, 10, and 17:

-state machine further includes a means for transitioning from said IF state to said END_IF state,

in response to recognizing an END_IF statement.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claims 4, 11, and 18:

-state machine further includes a means for transitioning from said IF state to said ELSE_IF

state, in response to recognizing an ELSE_IF clause.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claims 5, 12, and 19:

-state machine further includes a means for transitioning from said ELSE state to said END_IF

state, in response to recognizing an END_IF statement.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claims 6, 13, and 20:

-state machine further includes a means for transitioning from said ELSE_IF state to said

END_IF state, in response to recognizing an END_IF statement.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claims 7, 14, and 21:

-state machine further includes a means for transitioning from said ELSE_IF state to said ELSE

state, in response to recognizing an ELSE clause.

(As noted above, the means for transitioning from one state to another is via the controller

encountering a structured language term and referring to the label for executable code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claim 22:

22. An assembler residing in a data processing system for processing structured assembly

language, said assembler comprising:

-means for implementing a state machine having an IF state, and ELSE state, and END_IF state,

and ELSE_IF state, and a SETUP_IF state:

(Col. 2, lines 23-26, "...control system for a computer wherein a very small number of very high

level control constructs (IF, ELSE, ELSEIF, etc.)are capable of providing the entire control

structure of a very large system." Col. 2, line 34, "...programs and their labels are loaded...",

Col. 2, lines 39-45, "When a label for a set of instructions, constituting a specific program

segment, is detected, the label table is instructed to store, at the address represented by the label,

the address of the particular program segment which is derived from the program

counter...whenever it is desired to call out a new set of instructions, this is done by using the label as an address ..." Col. 2, lines 50-52, "Provision is made for conditional execution and branching as well as nesting, using labels in place of addresses." Conditional constructs direct execution to labels.)

-means for identifying a SETUP_IF clause;

(Mead failed to specify "SETUP_IF" as a clause. Official notice is given that a SETUP_IF clause is merely a label, defined by a programmer to define a location for an executing program to jump to, a control construct, when certain conditions are met. When the requirements on one state are met the program may be coded to jump to another following state.

Col. 2, lines 39-45, "When a label for a set of instructions, constituting a specific program segment, is detected, the label table is instructed to store, at the address represented by the label, the address of the particular program segment which is derived from the program counter...whenever it is desired to call out a new set of instructions, this is done by using the label as an address ..." Col. 2, lines 50-52, "Provision is made for conditional execution and branching as well as nesting, using labels in place of addresses." Labels are used to identify instructions for a control construct such as SET_UP. Although Mead did not disclose the SETUP_IF construct, it could be programmed by the developer. The conditional is stored when the program is scanned. A label relating to its executable code is created.)

-means for associating said identified SETUP_IF clause with an ELSE_IF clause having a test

condition;

(Col. 11, lines 49-50, "If during the execution of one procedure, another procedure call is

encountered, the process is repeated…" Col. 11, line 67 – col. 12, line 2, "…the processor

automatically threads its way backwards from procedures nested (associated constructs) as deep

as desired.)

-means for inserting instructions from said identified SETUP_IF clause prior to the test condition

of said ELSE_IF clause where said ELSE_IF clause logically follows a prior IF clause or a prior

ELSE_IF clause.

(Inserted executable code for the construct SETUP_IF will be referenced in the label table. A

test condition of ELSE_IF clause logically following a prior IF clause or a prior ELSE_IF clause

could determine whether to execute such code. Mead disclosed, "…the labels can be detected as

they are being transferred into the program memory (inserting instructions) …" As noted above,

the control process of a structured program can use various instructions (IF, ELSE, ELSE_IF,

etc.) to direct the flow of execution depending on a condition.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Per claim 23:

A computer program product residing on a computer usable medium for processing structured

assembly language, said computer program product comprising:

-program code means for implementing a state machine having an IF state, and ELSE state, and

END_IF state, and ELSE_IF state...

(Mead disclosed the IF and ENDIF instructions (col. 6, line 39-col. 7, line 4), the ELSE

instruction (col. 7, lines 5-34), the ELSEIF instruction (col. 7, line 35-col. 8, line 31) and nesting

(col. 9, lines 40-60). The state machine / micro program control capable of processing is

disclosed at col. 14, lines 22-33, "During the run mode, when the program control sees an IF

statement...the comparator compares...This condition is then written into the label stack

memory...along with the label corresponding to the IF statement. This condition is used by the

micro program control to determine whether or not the condition for a particular IF or ELSEIF

statement has been successfully met.")

Mead failed to disclose:

-a SETUP_IF clause;

-program code means for identifying a SETUP_IF clause;

-program code means for associating said identified SETUP_IF clause with an ELSE_IF clause

having a test condition;

-program code means for inserting instructions from said identified SETUP_IF clause prior to

the test condition of said ELSE_IF clause where said ELSE_IF clause logically follows a prior IF

clause or a prior ELSE_IF clause.

Official notice is given that a SETUP_IF clause is merely a label, defined by a programmer to

define a location for an executing program to jump to, a control construct, when certain

conditions are met. When the requirements on one state are met the program may be coded to

jump to another following state.

A test condition of ELSE_IF clause logically following a prior IF clause or a prior ELSE_IF

clause could determine whether to execute such code. Mead disclosed, "...the labels can be

detected as they are being transferred into the program memory (inserting instructions) ..." As

noted above, the control process of a structured program can use various instructions (IF, ELSE,

ELSE_IF, etc.) to direct the flow of execution depending on a condition.

Col. 2, lines 39-45, "When a label for a set of instructions, constituting a specific program

segment, is detected, the label table is instructed to store, at the address represented by the label,

the address of the particular program segment which is derived from the program

counter...whenever it is desired to call out a new set of instructions, this is done by using the

label as an address ..." Col. 2, lines 50-52, "Provision is made for conditional execution and

branching as well as nesting, using labels in place of addresses." Labels are used to identify

instructions for a control construct such as SET_UP. Although Mead did not disclose the

SETUP_IF construct, it could be programmed by the developer. The conditional is stored when

the program is scanned. A label relating to its executable code is created.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention, to have modified Mead's invention to include a SETUP_IF construct for the

purpose of directing control flow, because it merely is a name that references executable code, as

written by a programmer, for the purpose of logically organizing a program. Whatever the

control construct is named, depending on the result of the comparison, it references the label

table to find related code.

Response to Arguments

6.   Applicant has argued, in substance, the following:

(A) As Applicant has noted on page 6, 6th paragraph of Amendment received 17 August 2004,

"Mead does not teach or suggest "structured assembly language."

Examiner's Response:

In response to applicant's arguments, the recitation "structured assembly language" has

not been given patentable weight because the recitation occurs in the preamble. A preamble is

generally not accorded any patentable weight where it merely recites the purpose of a process or

the intended use of a structure, and where the body of the claim does not depend on the preamble

for completeness but, instead, the process steps or structural limitations are able to stand alone.

See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150,

152, 88 USPQ 478, 481 (CCPA 1951).

Structured assembly language is well known in the art. Note: US Patent 5,598,564 to

Barker, III (1994) Col. 7, lines 22, "the assembler will insert..." and line 27, "...directly

assembled...into machine language...", references to assembly language being assembled in

processing code. Col. 7, lines 4-11, "...a call to a label for a procedure to be performed while in

the state identified by the state label...call to a particular one of a plurality of tests that are

performed on current conditions...", "...parameters are state labels for other states to which the

state machine should branch on the N conditions..."

(A) As Applicant has noted on page 6, 7th paragraph of Amendment, "mead does not disclose the

claimed SETUP_IF state, Also, contrary to the characterization by the Examiner, the claimed

SET_UP state is not a label merely created by a programmer."

Examiner's Response:

Examiner disagrees. Assembly language controls flow using jump instructions. This is well

known in the art. See text book, "Mastering Turbo Assembler", by Tom Swan, Hayden Books,

1989, page 111, last paragraph, "All conditional jumps require a target address – **a label**

marking the location where you want the program to continue running if the specified

condition is met." (emphasis added) Meaning, if the state conditions are met, the program flows

to the next program state at an identified label. The label may be identified by whatever term the

programmer chooses. A SETUP_IF state is merely a label chosen by a programmer that contains

code to initialize.

Conclusion

7.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

       A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

8.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

       Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The

examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM   If

attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan

Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where

this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

11/22/2004

CHAMELI C. DAS
PRIMARY EXAMINER

11/26/04